

**МИНИСТЕРСТВО ПО РАЗВИТИЮ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
И КОММУНИКАЦИЙ РЕСПУБЛИКИ УЗБЕКИСТАН
ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

Факультет «Телекоммуникационные технологии»

**Кафедра «Технологии
мобильной связи»**

**ИСПОЛЬЗОВАНИЕ ПРОГРАММЫ МАТЛАВ
В УЧЕБНОМ ПРОЦЕССЕ**
по дисциплине «Мобильные системы связи»
методические указания

Ташкент 2016

ное средство решения инженерно-технических задач. Она является универсальным программным продуктом, позволяющим решать широкий спектр задач от символьных вычислений до сложных инженерных расчетов.

При разработке собственных настроек в системе Matlab для управления работы над специфическими задачами, а зачастую и просто при решении задач требуются определенные навыки, которые можно отнести к навыкам программирования. И хотя Matlab не является ни средой, ни языком программирования, она все же предоставляет собой базовые средства, присутствующие в большинстве языков программирования.

Данное методическое пособие позволит студентам познакомиться с этими основными средствами и приобрести навыки их использования.

Введение

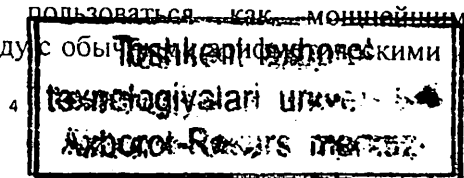
Система MatLAB создана фирмой MathWork Inc. (США, г. Нейтик, штат Массачусетс). Хотя впервые эта система начала использоваться в конце 70-х годов, расцвет ее применения начался в конце 80-х. Последние версии MatLAB, - это чрезвычайно развитые системы, которые содержат огромную совокупность процедур и функций, необходимых инженеру и научному работнику для осуществления сложных численных расчетов, моделирования поведения технических и физических систем, оформления результатов этих расчетов в наглядном виде.

Система MatLAB (сокращение от MATrix LABoratory - матричная лаборатория) представляет собой интерактивную компьютерную систему для выполнения инженерных и научных расчетов, ориентированную на работу с массивами данных. Система предполагает возможность обращения к программам, которые написаны на языках FORTRAN и C.

Привлекательной особенностью системы является то, что она содержит встроенную матричную и комплексную арифметику. Система поддерживает выполнение операций с векторами, матрицами и массивами данных, расчет ранга и чисел обусловленности матриц, поддерживает работу с алгебраическими полиномами, решение нелинейных уравнений и задач оптимизации, интегрирование функций в квадратурах, численное интегрирование дифференциальных и разностных уравнений, построение разнообразных видов графиков и т.д. В ней реализована удобная операционная среда, которая позволяет формулировать проблемы и получать решения в обычной математической форме.

Основной объект системы MatLAB - прямоугольный числовой массив (матрица), который допускает комплексные элементы. Использование матриц не требует явного указания их размеров. Система позволяет решать многие вычислительные задачи за значительно меньшее время, чем то, которое необходимо для написания соответствующих программ на языках FORTRAN, BASIC и C.

Система MatLAB выполняет операции с векторами и матрицами даже в режиме непосредственных вычислений без какого-либо программирования. Ею можно пользоваться как мощнейшим калькулятором, в котором наряду с обычными функциями



алгебраическими действиями могут использоваться такие сложные операции, как обращение матрицы, вычисление ее собственных значений и векторов, решение систем линейных алгебраических уравнений и многое другое. Тем не менее, характерная основная особенность системы – ее "открытость", то есть легкость ее модификации и адаптации к конкретным задачам пользователя. Пользователь может ввести в систему любую новую команду, оператор или функцию и пользоваться потом ими так же просто, как и встроенными операторами и функциями.

В базовый набор слов системы входят: спецзнаки; знаки арифметических и логических операций; арифметические, тригонометрические и некоторые специальные математические функции; функции быстрого преобразования Фурье и фильтрации; векторные и матричные функции; средства для работы с комплексными числами; операторы построения графиков в декартовой и полярной системах координат, трехмерных поверхностей и т.п.

Система MatLAB имеет собственный язык программирования. Запись программ в системе является традиционной и потому обычной для большинства пользователей персональных компьютеров. И вдобавок система дает возможность редактировать программы при помощи любого привычного для пользователя текстового редактора.

MatLAB имеет широкие возможности для работы с сигналами, для расчета и проектирования аналоговых и цифровых фильтров, для построения их частотных, импульсных и переходных характеристик. В наличии и средства для спектрального анализа и синтеза, в частности, для реализации прямого и обратного преобразования Фурье. Благодаря этому система довольно удобна для проектирования электронных устройств.

Работа в среде MatLab может осуществляться в двух режимах:

- в режиме калькулятора, когда вычисления осуществляются сразу после набора очередного оператора или команды MatLab;
- путем вызова имени программы, написанной на языке MatLAB, предварительно составленной и записанной на диске.

В обоих режимах пользователю доступны практически все вычислительные возможности системы, в том числе по выводу информации в графической форме. Программный режим позволяет сохранять разработанные вычислительные алгоритмы и, таким образом, повторять вычисления при входных данных.

Система MATLAB

Система MATLAB состоит из основных частей:

Язык MATLAB. Это язык матриц и массивов высокого уровня с управлением потоками, функциями, структурами данных, вводом-выводом и особенностями объектно-ориентированного программирования. Это позволяет как программировать в "небольшом масштабе" для быстрого создания черновых программ, так и в "большом" для создания больших и сложных приложений.

Среда MATLAB. Это набор инструментов и приспособлений, с которыми работает пользователь или программист MATLAB. Она включает в себя средства для управления переменными в рабочем пространстве MATLAB, вводом и выводом данных, а также создания, контроля и отладки М-файлов и приложений MATLAB.

Управляемая графика. Это графическая система MATLAB, которая включает в себя команды высокого уровня для визуализации двух- и трехмерных данных, обработки изображений, анимации и иллюстрированной графики.

Программный интерфейс. Это библиотека, которая позволяет писать программы на Си и Фортране, которые взаимодействуют с MATLAB. Она включает средства для вызова программ из MATLAB (динамическая связь), вызывая MATLAB как вычислительный инструмент и для чтения-записи файлов.

Описание среды системы MATLAB

Система MATLAB является интерактивной системой для выполнения инженерных и научных расчетов, ориентированной на работу с массивами данных.

В системе MATLAB можно выделить следующие основные функциональные элементы:

1. Командное окно для интерактивного ввода команд с клавиатуры.

Командное окно MATLAB

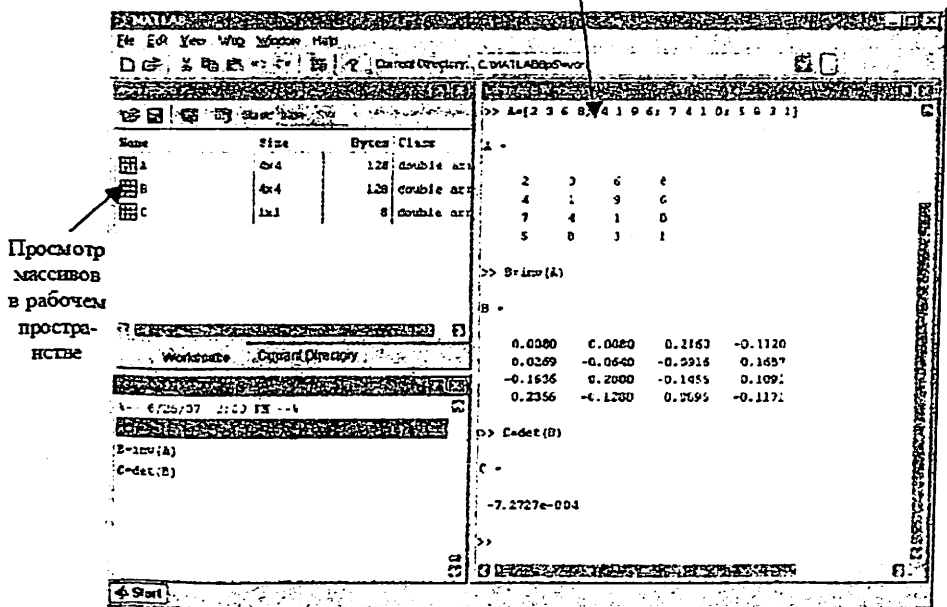


Рис. 1. Интерфейс среды MATLAB

2. Рабочее пространство, содержащее данные, с которыми производится работа. Данные представляются в виде переменных - массивов.

3. Библиотеки (Toolboxes) функций и команд MATLAB, включая стандартные команды MATLAB

4. Редактор m-файлов, предназначенный для создания и редактирования скриптов и функций MATLAB.

5. Справочная система MATLAB.

Простейшим способом работы с командами MATLAB является их ввод в командном окне и получение ответа в интерактивном режиме. Однако интерактивный ввод команд с клавиатуры не всегда удобен, поэтому предусмотрена возможность написания скриптов (сценариев).

Все переменные в системе MATLAB представляются в виде массивов.

Поэтому каждая переменная обладает следующими свойствами:

1. Размерность массива – количество измерений (для вектора размерность равна 1 – одномерный массив; для прямоугольной матрицы – 2 – двумерный массив).
2. Длина массива – количество элементов массива.
3. Размер массива по каждому измерению.
4. Тип данных элементов массива (таблица 1).

Скалярные переменные в системе MATLAB представляются как массивы единичной длины. При создании числовой переменной тип данных по умолчанию устанавливается в числовой с плавающей точкой (тип double). Переменные числового типа могут принимать кроме числовых значений значения Inf, -Inf и NaN, которые означают бесконечность, минус бесконечность и не числовое значение соответственно.

Таблица 1. Типы данных среды MATLAB

Тип	Описание
int8 int16 int32 int64	8, 16, 32, 64 – битные знаковые целые числа
uint8 uint16 uint32 uint64	8, 16, 32, 64 – битные без знаковые целые числа
single double	Числа с плавающей точкой с единичной и с двойной точностью
logical	Двоичные числа: 1 (истина), 0 (ложь)
char	Символьные переменные
cell	Массив ячеек разных типов данных

Также имеются зарезервированные переменные среды и управляющие функции MATLAB, которые представлены в таблицах 2 и 3.

Таблица 2. Зарезервированные переменные среды MATLAB

Имя переменной	Описание
<code>pi</code>	Содержит значения числа «пи»
<code>i, j</code>	Комплексная единица
<code>eps</code>	Относительная точность числовых значений с плавающей точкой
<code>ans</code>	Содержит результат выполнения последней команды
<code>%</code>	Символ, обозначающий начало комментария, вся строка после этого символа может содержать любой текст
<code>;</code>	Символ, обозначающий переход к следующему действию без отображения результатов команды в диалоговое окно
<code>Inf</code>	Указание бесконечности (например, возникает в результате деления на ноль)
<code>Nan</code>	Указание нечислового значения (например, неопределенность ноль деленный на ноль)

Таблица 3. Управляющие функции

Название функции или команды	Описание
<code>help</code>	Выводит описание функции или команды в командное окно
<code>lookfor</code>	Выводит список функций, в описании которых входит ключевое слово
<code>which</code>	Дает путь m-файла, в котором содержится указанная функция
<code>load</code>	Загружает переменные рабочей области MATLAB из файла *.mat
<code>save</code>	Сохраняет переменные рабочей области MATLAB в файл *.mat
<code>clear</code>	Удаляет переменные в рабочей области (пример: <code>clear all</code>)
<code>clc</code>	Отчищает командное окно
<code>echo</code>	Включает/выключает вывод результата, по умолчанию включен

Создание и работа с переменными

При создании переменных в системе MATLAB нет необходимости определять явно имя переменной, тип данных или размерность массива.

Когда MATLAB встречает новое имя переменной, автоматически создается переменная с таким именем и автоматически выделяется необходимое количество памяти.

Для этого могут использоваться скобки, которые осуществляют объединение скаляров или массивов, при этом в результате получается также массив (см. примеры в таблице 4).

Элементы внутри скобок могут быть разделены запятой, либо точкой запятой, в зависимости от того, по какому измерению осуществляется объединение. В системе MATLAB принято, что для прямоугольных матриц (двумерных массивов) первое измерение определяет число строк, а второе измерение – число столбцов. При использовании квадратных скобок запятая разделяет строки, а точка с запятой разделяет столбцы.

Задание массивов состоящих из диапазона равноотстоящих чисел может быть осуществлено с помощью символа двоеточия $a:b:c$. При этом указывается значение первого элемента массива a , шаг приращения b и максимально возможное значение (минимально возможное, если b отрицательно) в массиве.

Также существует ряд функций, которые позволяют создавать и заполнять массивы разного размера (см. таблицу 5).

Таблица 4. Примеры создания переменных

Команды	Описание
<code>NumVar = 0.0321;</code>	Команда создает числовую переменную с именем NumVar, и значением которой является число 0.0321
<code>Text = 'Текст';</code>	Команда создает символьную переменную с именем Text, и значением которой является строка 'Текст'
<code>A = [1; 2];</code>	Команда создает переменную с именем A, и значением которой является массив размером 2x1: 1 2

$B = [1, 2];$	Команда создает переменную с именем B , и значением которой является массив размером 1×2 : 1 2
$D = 1:5;$	Команда создает переменную с именем D , и значением которой является массив размером 1×5 : 1 2 3 4 5
$F = \text{zeros}(2, 3);$	Команда создает переменную с именем F , и значением которой является массив размером 2×3 , заполненный нулями

Таблица 5. Функции создания элементарных массивов

Название функции или команды	Описание
<code>eye</code>	Создает единичную матрицу, например, команда <code>eye(2)</code> создаст матрицу размером 2×2 : 1 0 0 1
<code>ones(m, n)</code>	Создание массива размером $m \times n$ заполненного единицами
<code>zeros(m, n)</code>	Создание массива размером $m \times n$ заполненного нулями
<code>rand(m, n)</code>	Создание массива размером $m \times n$ заполненного случайными числами, равномерно распределенными в интервале (0; 1)
<code>randn(m, n)</code>	Создание массива размером $m \times n$ заполненного случайными числами, нормально распределенными с нулевым средним и единичной дисперсией

Доступ к отдельным элементам массива осуществляется с помощью указания индексов в скобках. Индексы (номера) по разным измерениям отделяются запятой. Таким образом, если создан двухмерный массив с именем *Matrix*, то доступ к элементу массива, который находится на пересечении *m*-ой строки и *n*-ого столбца, осуществляется следующим образом:

$$Element = Matrix(m, n);$$

Доступ получить можно не только к отдельному элементу массива, но и к целому блоку. Для этого задаются не отдельные индексы, а массивы индексов. Например, команда

$$Block = Matrix([2, 3], [4, 5])$$

запишет в переменную *Block* квадратную матрицу, образованную из элементов массива *Matrix*, которые расположены на пересечении 2 и 3 строки и 4 и 5 столбца массива *Matrix*.

Кроме того, вместо индексов можно указывать специальные символы. Символ двоеточия указывает, что необходимо взять весь диапазон индексов по данному измерению. Символ *end* указывает, что необходимо взять последний элемент по данному измерению.

Например команда

$$Vector = Matrix(:, end)$$

запишет в переменную *Vector* последний столбец массива *Matrix*.

В таблице 6 приведены основные функции для определения параметров переменных.

Таблица 6. Основные функции для определения параметров переменных

Название функции или команды	Описание
<code>length(A)</code>	Определяет длину вектора A
<code>ndims(A)</code>	Определяет количество измерений массива A
<code>numel(A)</code>	Определяет количество элементов массива A
<code>size(A)</code>	Определяет размер массива A
<code>disp(A)</code>	Выводит значение переменной A в командное окно
<code>isscalar(A)</code>	Определяет, является ли переменная A скаляром

Математические выражения и функции

Большинство функций в системе MATLAB предназначены для работы с массивами. Поэтому математическим функциям, таким как `sin`, `log`, в системе MATLAB можно в качестве аргумента передавать массив числовых значений. При этом результат вычисляется для каждого элемента массива отдельно. Исключением являются специализированные для работы с матрицами функции, которые работают не поэлементно.

В таблицах 7 и 8 приведены операторы среды MATLAB и некоторые элементарные математические функции.

Таблица 7. Операторы среды MATLAB

Операторы	Описание
<code>()</code>	Задают порядок вычислений
<code>*</code> /	Операции умножения и деления
<code>^</code>	Возведение в степень
<code>`</code>	Операция транспонирования
<code>+</code> -	Сложение и вычитание
<code><</code> <code>></code>	Оператор меньше или больше
<code><=</code> <code>>=</code>	Оператор меньше или равно и больше или равно
<code>=</code> <code>~</code>	Оператор равно и неравно
<code>&</code>	Оператор «и»
<code> </code>	Оператор «или»
<code>~</code>	Оператор «не»

Таблица 8. Некоторые элементарные математические функции

Название функции или команды	Описание
<code>sin</code>	Вычисляет синус
<code>cos</code>	Вычисляет косинус
<code>tan</code>	Вычисляет тангенс
<code>atan</code>	Вычисляет арктангенс
<code>exp</code>	Возведение в степень
<code>log</code>	Натуральный логарифм
<code>sqrt</code>	Корень квадратный
<code>round</code>	Округление числа до целого

Графические команды и функции

Система MATLAB удобна для вывода данных в графическом виде. Для этого существует ряд высокоуровневых функций. Эти функции позволяют построить гистограммы, графики линий и поверхностей. При этом они автоматически выставляют параметры отображения. Но любые параметры отображения графиков могут быть изменены с помощью специальных низкоуровневых функций.

Для отображения графической информации создается отдельное окно. Такое окно может быть создано командой `figure`. Каждая высокоуровневая функция выводит графическую информацию в текущее графическое окно. При этом содержимое окна перезаписывается. Если такое окно не создано, то высокоуровневые функции автоматически создают его.

Низкоуровневые функции работают с текущим графическим окном. Они предназначены для изменения параметров отображения, а также для вывода отдельных графических элементов.

Таблица 9. Некоторые высокоуровневые графические функции

Название функции или команды	Описание
<code>plot</code>	Строит линии или точки в линейном масштабе
<code>loglog</code>	Строит линии или точки в логарифмическом масштабе
<code>polar</code>	Строит линии или точки в полярных координатах

Таблица 10. Некоторые низкоуровневые графические функции

Название функции или команды	Описание
<code>figure</code>	Графическое окно
<code>title</code>	Имя графика
<code>xlabel</code>	Метка по оси X
<code>axis</code>	Изменение масштаба по оси
<code>subplot</code>	Разделения окна на несколько областей

При работе с графическими функциями необходимо понимать объект - ориентированную структуру графики. Для вывода графических элементов создается ряд объектов:

1. Графическое окно (figure).
2. Система координат (axes).
3. Различные графические элементы (line, patch, text, surface, image, light).

Каждому из этих объектов ставится в соответствие идентификатор (дескриптор - handle). Знание идентификатора объекта, дает возможность читать и изменять свойства объекта с помощью функций get и set.

Рассмотрим пример использования введенных выше понятий для написания программы в MATLAB.

Пример использования программы MATLAB для формирования сигнала с использованием двоичной амплитудной модуляции

Программа формирования сигнала может иметь следующий вид:

```
% Задаем частоту дискретизации
SamplingRate = 1000;
% Определяем период дискретизации
SamplingPeriod = 1/SamplingRate;
% Задаем информационную последовательность битов
Bits = [1, 0, 0, 1, 1, 0, 1, 0, 1, 1];
% Определяем количество передаваемых битов
NoBits = length(Bits);
% Задаем длительность одного бита в секундах
TBit = 1;
% Определяем длительность всей последовательности битов в секундах
SignalDuration = T * NoBits;
% Формируем массив значений времени в секундах, длина массива
соответствует количеству дискрет в создаваемом сигнале
Time = 0 : SamplingPeriod : SignalDuration;
% Формируем сигнал последовательности битов, используя
функцию интерполяции
SignalBits = interp1(0.5 : NoBits, Bits, Time, 'nearest', 'extrap');
```

```

% Задаем период несущего синусоидального сигнала в секундах
TSin = 0.1;
% Определяем частоту несущего синусоидального сигнала в герцах
SinRate = 1/TSin;
% Формируем несущий синусоидальный сигнал
SignalSin = sin( 2 * pi * SinRate * Time );
% Модулируем несущий синусоидальный сигнал сигналом
последовательности бит
ModulatedSignal = SignalBits .* SignalSin;
close all % закрываем окна для рисования открытые ранее

```

```

% Рисуем график сигнала последовательности бит
H = figure; % Создаем окно для рисования
set(gcf,'Units', 'normalized'); % Определяем
set(H, 'Position', [0.1 0.63 0.8 0.25]);% положения окна
plot(Time, SignalBits); % Рисуем сигнал
axis([0 SignalDuration -0.5 1.5]); % Изменяем масштаб отображения
title('Bits signal'); % Задаем название графика
xlabel('Time, seconds'); % Задаем метку временной оси

```

```

% Рисуем график синусоидального сигнала
H = figure; % Создаем окно для рисования
set(gcf,'Units', 'normalized'); % Определяем
set(H, 'Position', [0.1 0.33 0.8 0.25]);% положения окна
plot(Time, SignalSin); % Рисуем сигнал
axis([0 SignalDuration -1.5 1.5]); % Изменяем масштаб отображения
title('Sinusoidal signal'); % Задаем название графика
xlabel('Time, seconds'); % Задаем метку временной оси

```

```

% Рисуем график модулированного сигнала
H = figure; % Создаем окно для рисования
set(gcf,'Units', 'normalized'); % Определяем
set(H, 'Position', [0.1 0.03 0.8 0.25]);% положения окна
plot(Time, ModulatedSignal); % Рисуем сигнал
axis([0 SignalDuration -1.5 1.5]); % Изменяем масштаб отображен
title('Modulated signal'); % Задаем название графика
xlabel('Time, seconds'); % Задаем метку временной оси

```

Результаты графического отображения последовательности бит, синусоидального и модулированного сигналов представлены на рисунках 2,3 и 4.

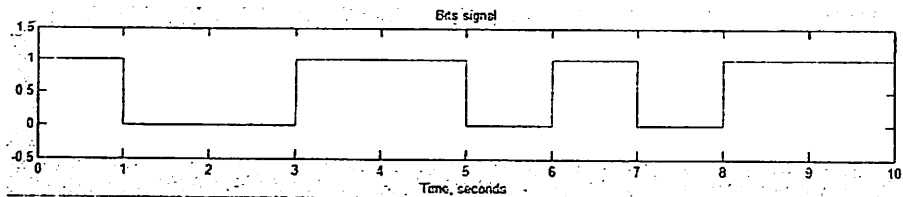


Рис.2. Графическое отображение сигнала последовательности бит

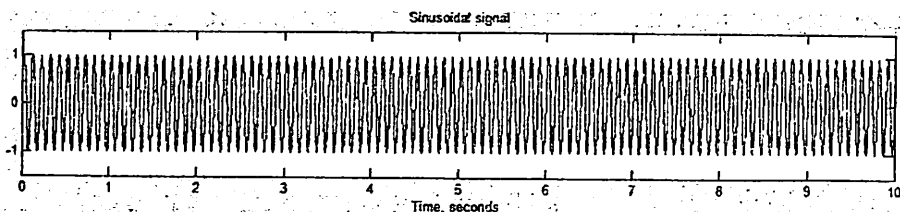


Рис.3. Результат графического отображения синусоидального сигнала

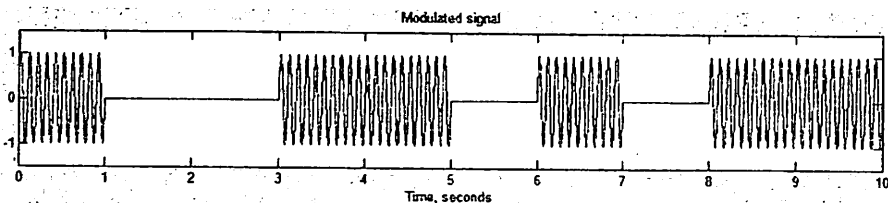


Рис.4. Результат графического отображения модулированного сигнала

В качестве заключения можно отметить, что Matlab является удобным средством по моделированию и визуализации проектируемых объектов, и может использоваться в учебном процессе.

Литература

1. Максимей И.В. Имитационное моделирование на ЭВМ.-М.: Радио и связь. 1988 г.
2. Лузина Л.И. Компьютерное моделирование. Учебное пособие. Томск 2001 г.
3. Ануфриев И, Смирнов А. Matlab 7. Полное руководство. Санкт-Петербург, 2005 г.
4. Васильев В.В. Математическое и компьютерное моделирование процессов и систем. Учебное пособие. Украина, 2008 г.

ИСПОЛЬЗОВАНИЕ ПРОГРАММЫ МАТЛАБ В УЧЕБНОМ ПРОЦЕССЕ

по дисциплине «Мобильные системы связи»
методические указания

Работа рассмотрена на факультете
«Телекоммуникационных технологий»
(протокол №9 25.05.16).

Работа рассмотрена и одобрена на заседании
кафедры ТМС (протокол №8 от 5.05.16).

Составитель ст.преп. Хатамов А.П.

Редактор Ибраимов Р.Р.

Корректор Хамдам –Зода Л.Х.

Бумага офсетная. Заказ № 6.2

Тираж 50

Отпечатано в типографии ТУИТ

Ташкент 700084, ул.А.Тимура - 108